Amendments to the Specification:

For all of the following changes to both the specification and the claims, subject matter to be removed is shown with a ~~strikeout~~, and subject matter to be added is shown as <u>underlined</u>.

Please replace the paragraph at page 4, lines 10-18 with the following:

Typical solutions for the data ingest problem are often hard coded for a single purpose and there is no reusability of the software. Additionally, slight variation in data can cause significant problems to solutions that are hard coded for a single purpose. In many instances commercial parsing software for reformatting data does not exist. Thus, the burden is on the user to re-format the desired data. Alternatively, the user may hire a programmer to generate or modify software that can reformat the desired data. Both of these solutions are relatively expensive because they take time to implement and require user resources, require developer time, and give the user a solution~~,~~ which the user can not modify.

Please replace the 3 paragraphs appearing on page 5 with the following:

A method for extracting ~~a plurality of structured~~ <u>and converting</u> data from one or more information sources <u>into a common format</u>. The method comprises receiving the information sources, receiving at least one pattern descriptor selected from a graphical user interface, and receiving one or more templates with each template having at least one pattern descriptor. The method then proceeds to apply the one or more templates to the information sources. The method generates the plurality of ~~structured~~ data in a common format by parsing the information sources with the templates. The method stores the ~~structured~~ data in the common format.

A system for extracting ~~a plurality of structured~~ <u>and converting</u> data from one or more information sources <u>into a common format</u> is also disclosed. The system includes a

memory, an input device and a processor. The memory is configured to receive the information sources and store the templates. The input device is configured to receive the pattern descriptors from a user interacting with the graphical user interface. The processor is programmed to apply the templates to the information sources, to generate ~~structured~~ data in a common format by parsing the plurality of information sources with the templates, and communicate the ~~structured~~ data in the common format for storage.

The graphical user interface comprises a first button that permits the user to receive the information sources, a second button that permits the user to select a pattern descriptor, a third button that permits a user to select one or more templates, and a display window configured to display the ~~structured~~ data in the common format.

On page 6, please replace the sentence at line 12-13 with the following:

FIG. 9 is a screenshot of the GUI in which ~~structured~~ data having a common format is displayed.

On page 8, please replace the paragraph at lines 10-15 with the following:

Furthermore, those skilled in the art having the benefit of this disclosure shall appreciate that these illustrative systems and methods can be applied to a variety of applications that ~~are~~ require parsing information sources and generating a ~~structured~~ data output which has a common format. Further still, the illustrative embodiment describes an illustrative graphical user interface (GUI) ~~interface~~ for parsing information sources.

On page 10, please replace the two paragraphs at lines 5-17 with the following:

In the illustrative system, the memory is configured to receive the plurality of information sources and to store the templates that are used to generate the ~~structured~~ data <u>having a common format</u>. For the illustrative embodiment, the structured and semi-structured information sources comprise text data that is configured in a variety of

different formats. The systems and methods then parse the structured information sources and semi-structured information sources using templates. The templates may be stored in a template library or may be generated for a particular group of text documents.

After parsing the information sources, a plurality of ~~structure~~ data is generated in which the content is organized, ordered and grouped according to a plurality of pattern descriptors. The ~~structured~~ data is stored in a common format, which in the illustrative example is an extensible markup language (XML) format. As described in further detail below, the ~~structured~~ data having a common format can be stored in a storage bin such as an input bin, a wait bin, an incomplete bin, and a complete bin.

On page 11, please replace the two paragraphs at lines 8-20 with the following:

An illustrative user employs the pattern descriptor to generate templates which enable the parsing of ~~structured~~ data from information sources without having the user program or understand the algorithms used to perform the reformatting of the information sources. By way of example and not of limitation, the illustrative user may be an information analyst. In another illustrative example, the user may be a system integrator or operations analyst.

The processor 12 is programmed to apply the templates to the information sources. A plurality of ~~structured~~ data having a common format is generated by parsing the information sources with one or more templates. In the illustrative example described in further detail below, the generated ~~structured~~ data <u>having a common format</u> is stored as a text file. The processor then proceeds to communicate the generated ~~structured~~ data <u>having a common format</u> to an application configured to receive ~~structured~~ data having the common format. By way of example and not of limitation, the application is a database application.

On page 12, please replace the paragraph at lines 11-21 with the following:

In the illustrative client-server system 50, the client 54 has enabled a web browser that downloads a Java applet from server 56. The downloaded Java applet displays the GUI that is described in further detail below. The client 54 is in communication with server 56, which for this illustrative embodiment is a web server configured to use TCP/IP communication protocols. The web server is configured to host a number of programs such as Java servlet, Java applets, configuration files and other such files. In the illustrative client-server system 50, the server 56 is configured to parse information sources and generate ~~structured~~ data having a common format. The server 56 then proceeds to communicate the ~~structured~~ data <u>having a common format</u> to a file server (not shown), which stores the ~~structured~~ data <u>having a common format</u> as a text file. Those skilled in the art shall appreciate that the parsing can be performed in batches or on a real-time basis.

Please replace the paragraph on page 15, lines 5-13 with the following:

After receiving the information sources and the user selected template, the universal parsing agent 108 parses the text documents and generates the ~~structure~~ data having a common format 110. The ~~structured~~ data <u>having a common format</u> 110 is organized, ordered and grouped according to the template. The ~~structured~~ data is stored in a common format, which in the illustrative example is an extensible markup language (XML) format. The plurality of ~~structured~~ data is configured in a common format that can be used to automatically populate an application 112 such as a database 114. Those skilled in the art shall appreciate that a plurality of applications and databases may be populated with the ~~structured~~ data <u>having a common format</u> that was generated from the universal parsing agent.

On page 16, please replace the paragraph at lines 11-19 with the following:

At block 158, the method applies one or more templates to the information sources and generates a plurality of ~~structured~~ data in a common format such as XML. In the illustrative embodiment, each template is comprised of an XML schema that is defined by the user with pattern descriptors. Schemas define the characteristics of classes of objects. For example in Standard Generalized Markup Language (SGML) terminology, a text document has a document type and the formal definition that describes each document type is referred to as a document type definition (DTD). Thus, the DTD defines a set of valid tags for a document using standardized semantics and language.

On page 17, please replace the paragraph at lines 7-12 with the following:

At block 160, the plurality of ~~structured~~ data <u>having a common format</u> is generated by parsing the information sources with one or more templates. The parsing of the information sources can be performed in any natural language such as Chinese, Japanese, French, and English at one time. The parsing of the information sources may be performed without having to replicate the parsing process. Thus, the parsing process is not repeated for each natural language.

Please replace the paragraph beginning on page 17, line 18 and ending on page 18, line 5 with the following:

If the user decides not to modify the template, the method proceeds to block 166 where the ~~structured~~ data is stored in a common format. In the illustrative example, the ~~structured~~ data <u>having a common format</u> may be stored in one of four possible storage bins that comprise an input bin, a wait bin, an incomplete bin and a complete bin. The storage options are described in further detail below. In the illustrative example, the ~~structured~~ data <u>having a common format</u> is stored as a text file on a file server. The user then has the opportunity to communicate the ~~structured~~ data <u>having a common format</u> to an application that is configured to receive data having the common format. For

illustrative purposes only, the application may be relational database application or other database application.

Please replace the paragraph on page 19, lines 7-17 with the following:

At block 264, the ~~structured~~ data having a common format is generated using the method described above. The generated ~~structured~~ data <u>having a common format</u> can then be stored in the wait bin, the incomplete bin, or a complete bin. The waiting bin permits the user to view files that matched required items in a template, thereby permitting the user to manually revise the pattern descriptor for a modified template or to designate the file as complete. The incomplete bin lists all files where no direct matches were found with the available templates. For files in the incomplete bin, the user views these files and creates templates to parse these "incomplete" files, and uses new templates to reprocess any failed files. The complete bin lists files that have been successfully parsed and the template that was used to parse it. Additionally, for each storage bin the user has the ability to generate statistical information.

Please replace the paragraphs beginning on page 21, line 20 and ending on page 22, line 4 with the following:

Referring to FIG. 9, there is shown a screenshot of the GUI 350 in which ~~structured~~ data having a common XML format is displayed. In this illustrative example, the selected document "book2.txt" has been parsed. The display window 352 shows a plurality of structured data in an XML format. Another window 354 displays the pattern descriptors that are associated with the ~~structured~~ data file. Yet another window 356 shows the value associated with a particular tag 358.

Please replace the paragraph on page 22, lines 5-10 with the following:

It shall be appreciated by those skilled in the art having the benefit of this disclosure that the illustrative systems and methods described above have been developed to receive a plurality of information sources that are inconsistently formatted. The universal parsing agent proceeds to apply a user-defined template to generate ~~structured~~ data configured in a common format that can be used to automatically populate an application such as a database.

On page 31, please replace the abstract with the following:

A system and method for extracting ~~a plurality of structured~~ and converting data from one or more information sources into a common format. The method comprises receiving the information sources, receiving at least one pattern descriptor selected from a graphical user interface, and receiving one or more templates with each templates having at least one pattern descriptor. The method then proceeds to apply the one or more templates to the information sources. The method generates the plurality of ~~structured~~ data in a common format by parsing the information sources with the templates. The method stores the ~~structured~~ data in the common format.